自适应的邻域粗糙集邻域大小取值方法 *

彭潇然 [a]，刘遵仁 [b]，纪 俊 [b]

(青岛大学 a. 数据科学与软件工程学院; b. 计算机科学技术学院，山东 青岛 266071)

摘 要：邻域粗糙集应用的好坏依赖于邻域大小 δ 的取值。在使用基于邻域粗糙集的属性约简算法时，现有的 δ 取值方法一般是点值式的，即仅凭借人的经验指定某个值，这种方法在对 δ 取值时没有结合实际问题的具体情况，因此在算法的实用性上可以作进一步讨论。为此，提出一种自适应 δ 取值方法，其最大特点是不指定 δ 取值，而是指定 δ 取值的区间，然后在该取值区间上，通过使用一种结合了数据集和分类器自身特性的适应值函数自动地选出最合适的 δ 取值。实验结果表明，相比点值式 δ 取值方法，通过自适应 δ 取值方法能找到属性个数更少，而分类精度更高的属性集。实验证明该方法能进一步提高基于邻域粗糙集的属性约简算法的实用性。

关键词：邻域粗糙集；邻域大小；属性约简；分类

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2017.07.0676

Adaptable method for determining neighborhood size of neighborhood rough set

Peng Xiaoran[A], Liu Zunren[B], Ji Jun[B]

(*a. College of Data Science & Software Engineering, b. College of Computer Science & Technology, Qingdao University, Qingdao Shandong* 266071, *China*)

Abstract: The application of neighborhood rough set depends on the value of neighborhood size δ. When using attribute reduction algorithms based on neighborhood rough set, an existing method for determining δ is usually point-type, that is, to specify a value only by human experience. The method does not combine with the actual situation when it is used to determine δ, so the practicability of the algorithms can be further discussed. For this reason, an adaptable method for determining δ is proposed. The biggest characteristic is not determining δ but the interval of δ, then the most appropriate δ in the interval is forwardly selected by using a fitness function that is combined with the characteristics of data sets and classifiers. The experimental results show that, compared with the point -type method for determining δ, this method can find reduction sets which number of attributes is less, and classification accuracy is higher, which proves that this method can further improve the practicability of attribute reduction algorithms based on neighborhood rough set.

Key Words: neighborhood rough set; neighborhood size; attribute reduction; classification

0 引言

粗糙集理论认为知识是有粒度的，它是一种对论域中对象进行分类的能力[1]。经典的 Pawlak 粗糙集[2]采用等价划分和等价类的概念保证了粒度计算的进行，但是这种处理方式只适用于离散型变量，而现实应用中需要处理的数据类型往往是数值型的，这种局限滞缓了粗糙集理论的应用。

针对上述问题，Zadeh[3]提出了信息粒化和粒度计算的概念。Lin[4]在信息粒化、粒度的基础上提出了邻域模型的概念。Hu[5]基于邻域粒化和粗糙逼近，提出的邻域粗糙模型[6-7]可以处理数值型数据，这进一步拓展了粗糙集理论的应用范围。但是，和经典的 Pawlak 粗糙集不同，因为引入了邻域粒化的概念，基于邻域粗糙集的属性约简算法在对一个数据集进行属性约简时，其结果的好坏依赖于邻域大小 δ 的取值[5,8~16]。合适的 δ 取值能让算法得到较好的属性约简集合，而不合适的 δ 取值则会让算法得到一般，甚至是很差的属性约简集合。对于约简算法得到的约简集合，本文希望属性个数较少且有效性较高。其中，约简集合的有效性体现在分类器根据该约简集合对数据集进行分类后得到的分类精度上，分类精度越高，则约简集合的有效性越高。因此，针对 δ 取值的讨论就显得十分重要且有意义了。

现有的 δ 取值方法一般是点值式的，即仅根据人的经验指定 δ 为某个值[5, 9~16]。例如，Hu[5]在与其他算法的对比实验中，指定提出的邻域粗糙集算法中 $\delta = 0.125$；刘[9]在与其他算法的对比实验中，指定提出的邻域粗糙集算法中 δ 为数据集归一化

基金项目：国家自然科学基金资助项目（61503208）

作者简介：彭潇然（1994-），男，湖北天门人，硕士研究生，主要研究方向为粗糙集理论（pxr1203@qq.com）；刘遵仁（1963-），男，副教授，博士，主要研究方向为粗糙集理论、数据挖掘、智能计算等；纪俊（1982-），男，副教授，博士，主要研究方向为数据挖掘、大数据技术、转化医学等.

之后的标准差；段[11]在与其他算法的对比实验中，针对不同数据集，指定提出的邻域粗糙集算法中 δ 为不同值；Chen[15]在与其他算法的对比实验中，指定提出的邻域粗糙集算法中 $\delta=0.1$ 等等。但是，需注意到，对于不同的数据集和分类器，它们本身的性质是不同的。例如，有的数据集含有的噪声数据较少，而有的数据集则较多；同一数据集的某个属性集合可能在一个分类器上分类效果较好，而在另外一个分类器上的分类效果较差。因此，这种仅凭借经验，没有结合具体实际情况的 δ 取值方法，在一定程度上限制了属性约简算法的实用性。

针对上述问题，本文提出一种自适应的邻域粗糙集 δ 取值方法。该方法不指定 δ 取值，而是指定 δ 取值范围，通过结合数据集和分类器本身的性质，设计一种适应值函数用于评价取值范围中各 δ 取值的好坏，并自动选取适应值最大的 δ 取值作为最终结果。此外，可以通过调整适应值函数的权值使算法更好地满足实际应用的需求，进一步提升算法的实用性。

1　相关概念[2]

1.1　邻域粒化

定义 1　度量计算。给定 n 维实数空间 R^n，对于空间中的任意两个点 $x_i=(x_{i1},x_{i2}\ldots,x_{in})$ 和 $x_j=(x_{j1},x_{j2}\ldots,x_{jn})$，定义 $d(x_i,x_j)$ 是 R^n 上的一个度量计算，一般选用欧氏距离，即

$$d(x_i,x_j)=\left(\sum_{p=1}^{n}\left|x_{ip}-x_{jp}\right|^2\right)^{\frac{1}{2}}$$

定义 2　邻域粒子。在实数空间上，定义样本的非空有限集合 $U=\{x_1,x_2\ldots,x_n\}$，且称 U 为论域。定义 U 上的样本 x_i 的 δ-邻域为 $\delta(x_i)=\{x_j\mid x_j\in U,d(x_i,x_j)\le\delta\}$，其中 $\delta\ge 0$。$\delta(x_i)$ 称做由 x_i 生成的 δ-邻域信息粒子，简称为 x_i 的邻域粒子。

1.2　决策表及上下近似

定义 3　决策表。定义四元组 $NDT=(U,C\cup D,V,f)$ 为决策表。其中 U 是论域；C 是条件属性集；D 是决策属性集，且 $C\cap D=\varnothing$，$C\ne\varnothing$，$D\ne\varnothing$；V 是信息函数 f 的值域。

定义 4　上下近似。给定一个决策表 $NDT=(U,C\cup D,V,f)$，D 将 U 划分为 N 个等价类：$D_1,D_2,\ldots,D_N$，$\forall B\in C$，定义决策属性集 D 关于 B 的下近似和上近似为

$$\underline{N_B D}=\bigcup_{i=1}^{N}\underline{N_B D_i},$$

$$\overline{N_B D}=\bigcup_{i=1}^{N}\overline{N_B D_i}$$

其中，$\underline{N_B D_i}=\{x_i\mid\delta_B(x_i)\subseteq D_i,x_i\in U\}$，

$\overline{N_B D_i}=\{x_i\mid\delta_B(x_i)\cap D_i\ne\varnothing,x_i\in U\}$

定义决策属性集 D 关于 B 的正域为 $Pos_B(D)=\underline{N_B D}$，边界域为 $BND_B(D)=\overline{N_B D}-\underline{N_B D}$，负域为 $NEG_B(D)=U-\overline{N_B D}$。

2　F2HARNRS 算法及 δ 取值分析

为了分析 δ 取值对基于邻域粗糙集的属性约简算法的影响，首先介绍一种属性约简算法，然后基于此算法对不同的 δ 取值进行可靠性分析。

定义 6[5]　属性约简。给定一个决策表 $NDT=(U,C\cup D,V,f)$，$\forall B\subseteq C$，若满足 $\mathrm{Pos}_B(D)=\mathrm{Pos}_C(D)$，则称 B 是一个独立属性子集；如果对 $\forall a\in B$，$\mathrm{Pos}_{B-\{a\}}(D)<\mathrm{Pos}_B(D)$，则称 B 为 C 的一个属性约简。

贪心策略具有以较少时间求解最优解或次优解的特点。Hu[5]结合贪心思想构造了一种前向贪心的决策表属性约简算法 (Fast forward heterogeneous attribute reduction based on neighborhood rough sets, F2HARNRS)，F2HARNRS 算法被提出后得到了广泛地应用与研究[8-11]。本文中对 δ 取值的讨论将建立在该算法的基础上。

F2HARNRS 算法将原属性集下的正域样本个数作为贪心目标，其具体策略是：初始化属性约简集合为空集，此时当前正域为空集，每次选取使当前正域中样本个数增加最多的属性加入集合，直至待选的任意属性均不再增加正域中样本个数或待检验的样本全划入当前正域中时，输出集合。其中，根据新增加的属性不会使已属于正域的样本变为非正域样本这一性质，在算法的计算过程中，每次仅对还未判定为正域的样本进行正域计算。如算法 1 所示。

算法 1：

Input：决策表 $(U,C\cup D,V,f)$，δ

Output：属性约简集合 red

初始化 red=$\varnothing$，待检验样本 smp_chk=U，

样本数最多的正域集合 max_pos=$\varnothing$，max_pos=$\varnothing$ 对应的属性 max_i=$\varnothing$

while　smp_chk $\ne\varnothing$

　　max_pos $=\varnothing$；

for each　$k_i\in(C-\mathrm{red})$

　　$\mathrm{Pos}_i=\mathrm{Pos}(\mathrm{smp_chk},\mathrm{red}\cup k_i,D,\delta)$；

if　$|\mathrm{max_pos}|<|\mathrm{Pos}_i|$

　　max_pos=Pos_i；

　　max_i=k_i；

end if

end for

if　max_pos $\ne\varnothing$

　　red=red $\cup$ max_i；

　　smp_chk=smp_chk-max_pos；

else

break;

end if

end while

return red;

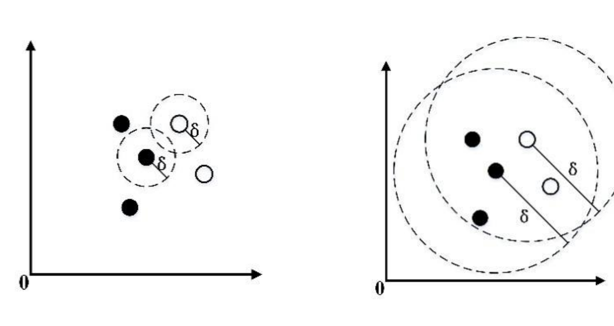
在算法 1 中，$\mathrm{Pos}(\mathrm{smp_chk},\mathrm{red}\cup k_i,D,\delta)$ 是正域计算函数，也是算法时间开销最大的部分。它的功能是：在当前 δ 取值下，求得 smp_chk 中决策属性 D 关于条件属性集合 red $\cup\, k_i$ 的正域。

在该算法下，假设某一数据集有 m 个属性，约简结果中包含 k 个属性，且每增加一个属性正域中增加 $\frac{|U|}{k}$ 个样本，则算法 1 进行样本是否属于正域的判定次数为

$$m|U|+(m-1)|U|\frac{k-1}{k}+\ldots+(m-k)|U|\frac{1}{k} < \frac{m|U|(1+2+\ldots+k)}{k} = \frac{m|U|(1+k)}{2}$$

可以看出，F2HARNRS 算法的效果与 δ 取值紧密相关，δ 取值不同则 F2HARNRS 算法的效果不同。这是因为在不同的 δ 取值下形成的邻域粒子 $\delta(x_i)$ 不同，这造成正域计算的结果不同，进而造成 F2HARNRS 算法根据正域计算得到的属性约简集合也不同。其中，不同约简集合的可靠性也是不同的。约简集合的可靠性主要体现在约简集合中属性个数和分类算法根据该约简集合进行分类后得到的分类精度上，即对原属性集是否达到了较好的约简效果，约简后是否仍能保证分类精度没有严重丢失。其次，F2HARNRS 算法的终止条件有 2 个：$\text{smp_chk}=\varnothing$ 和 $\text{max_pos}=\varnothing$，即待检验的样本全划入当前正域中或待选的任意属性均不再增加正域中样本个数，任何一个条件成立则算法结束。

分析在 δ 取值逐渐增大的过程中邻域粒子 $\delta(x_i)$ 的变化情况及对应约简集合的可靠性。如图 1 所示，图 1(a)中，在当前 2 维实数空间中，当 δ 取值较小时，$\delta(x_i)$ 只包含样本 x_i，此时原属性集中的任意两个条件属性都能使所有的待检测样本划入正域中，即使算法 1 中的条件 $\text{smp_chk}=\varnothing$ 成立，此时所得的属性约简没有意义，约简集合 red 的可靠性为零；图 1(b)中，随着 δ 取值的增大，$\delta(x_i)$ 中样本增多，直至在某个较大的 δ 取值下，$\delta(x_i)$ 包含整个论域中的样本，此时任意 2 个条件属性都不能使任意的待检测样本划入正域中，即使算法 1 中的条件 $\text{max_pos}=\varnothing$ 成立，此时正域中样本个数为 0，约简集合 red 为空，约简集合 red 的可靠性也为零。这说明随着 δ 取值的增大，约简集合 red 的可靠性先增大后减小。其中，这种可靠性的变化过程因数据集和分类器的不同而不同，因此仅凭经验的点值式 δ 取值方法在一定程度上并不能得到最好的 δ 取值和保证算法效果的最大化，限制了算法的实用性。



(a) δ 取值较小时的情况　　(b) δ 取值较大时的情况

图 1　δ 取值的两种极端情况

3　自适应的 δ 取值方法

根据第 2 节的分析，本文提出的 δ 取值方法的主要工作在于：将评价标准由仅依据人的经验改进为结合实际数据集和分类器的性质，且将评价 δ 取值的好坏转换为分析其对应约简集合可靠性的高低。其中，约简集合可靠性的高低可以通过数据直观地体现出来。自适应的 δ 取值方法的主要思想是：指定 δ 取值范围，并在该范围上选取多个待比较的 δ 取值，然后对各 δ 取值进行评价，最后将最好的 δ 取值作为结果。

结合实际情况，提出一种用于评价 δ 取值好坏的适应值函数，该函数主要包含两个变量 $length$ 和 $ratio$，表示对应约简集合的可靠性。其中，$length$ 表示在该 δ 取值下，属性约简算法所得的约简集合中属性个数的适应值；$ratio$ 表示在该 δ 取值下，分类器根据约简集合对数据集进行分类后所得的分类精度的适应值。δ 取值的适应值取决于这两个变量的加权求和，如式(1)所示。

$$fit(\delta)=\alpha\cdot length+\beta\cdot ratio \tag{1}$$

其中：α 和 β 是两个权值，且满足 $\alpha>0$，$\beta>0$，$\alpha+\beta=1$。权值的设置可以根据应用需求进行调整，增大侧重项相应权值的比重。本文实验部分采用 α=0.4，β=0.6。

本文 δ 取值方法的步骤为：

a)指定取值区间 $[a,b](a\neq 0)$，并选取 n 个待比较的 δ 取值；

b)计算各 δ 取值下属性约简算法得到的约简集合，并统计对应的属性个数和分类精度；

c)对上述两组记录 $length$ 和 $ratio$ 进行归一化处理，之后调整 $length=1-length$。即对某个约简集合而言，属性个数越少 $length$ 值越高，分类精度越高 $ratio$ 值越高；

d)代入式(1)中计算各 δ 取值的适应值 $fit(\delta)$，并选取适应值最大的 δ 取值作为最后结果。

根据第 2 节中的分析可知，约简集合 red 的可靠性是先增大后减小的，这说明 δ 取值过小或过大都是无意义的。一般来说取值区间 $[a,b](a\neq 0)$ 应包含于 $(0,0.8]$，可根据具体情况进行选取。在 Hu[5]的实验结论中，针对其选取的数据集和分类器，$[0.1,0.3]$ 是 δ 较好的取值区间，大多数分类器在此 δ 的取值下可以获得良好的分类性能。在本文实验中，参考 Hu[5]的实验结论，指定的取值区间为 $[0.02,0.4]$，以 0.02 增进，共取得 20 个待比较的 δ 取值，效果较好。合理的取值区间能避免无用功，提高该方法的效率。

其次，该方法的时间开销主要在于步骤 b)。若需要比较的 δ 取值个数为 n，属性约简算法的时间开销为 $O_{red}(\ )$，则该方法的时间开销可以表示成 $n\cdot O_{red}(\ )$。对于不同的属性约简算法，若其 $O_{red}(\ )$ 很大，可适当调整取值范围及增进量。

4　实验分析

在本次实验中，首先采用“F2HARNRS+SVM”的方式并通过三个维数不同的 UCI 数据集分析 δ 取值与属性个数及分类精

度之间的关系；然后对比点值式和自适应的 δ 取值方法，分别记录在这两种方法下，采用“F2HARNRS+SVM”方式和“F2HARNRS+1-NN”方式在 7 个不同的 UCI 数据集上所得的属性个数和分类精度；最后，分析实验结果并得出结论。其中，对于两种分类算法，随机地选取数据集中每类对象的 2/3 作为训练集，1/3 作为测试集，算法执行 20 次，分类精度的结果取均值。

4.1 实验环境

UCI(University of California Irvine)(http://archive.ics.uci.edu/ml/)提供了一系列用于测试的标准数据集。本文从 UCI 数据集中挑选了 7 个数值型数据集，其中，每个数据集提供了条件属性和决策属性。

表 1 数据集描述

数据集	样本数	属性数	类别数
Wine	178	13	3
WDBC	569	30	2
Sonar	208	60	2
WPBC	198	33	2
Ionosphere	351	33	2
Credit Approval	690	13	2
German Credit	1000	24	2

本次实验在一台 Intel(R) CoreTM i5 CPU 和 4 GB 内存的 PC 机上，采用 Windows 7 环境下的 MATLAB R2016b 进行算法仿真。

4.2 δ 取值与实验结果

4.2.1 取值和属性约简个数的关系

在区间 $[0.04,1]$ 上，按 0.04 增进，共取得 25 个 δ 取值，记录不同 δ 取值下 F2HARNRS 算法在数据集 Wine、WDBC 和 Sonar 上所得约简集合中属性的个数以及 SVM 分类算法根据该约简集合对数据集分类后的分类精度。

δ 取值与属性个数的关系如图 2 所示。

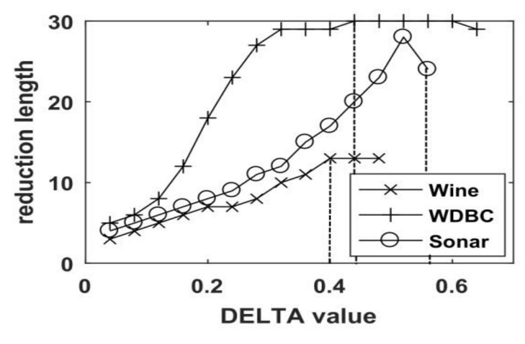
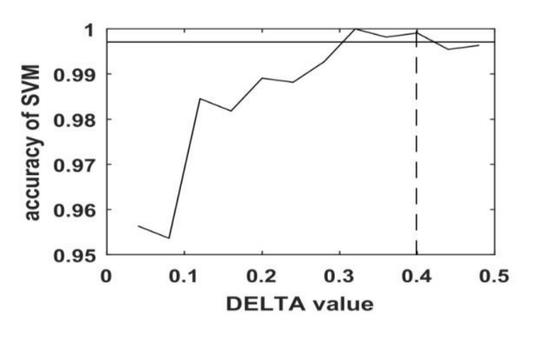


图 2 取值与属性个数之间的关系

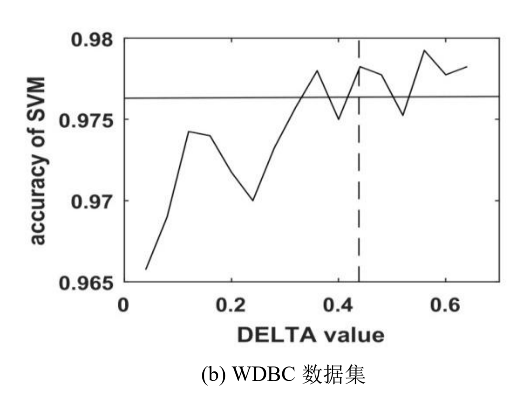
如图 2 所示，对同一数据集，δ 取值不同时，所得的属性个数也不同，过大或过小的 δ 取值没有对应的结果，这进一步说明过大或过小的 δ 取值均是无意义的。随着 δ 取值的增大，所得属性的个数增大，直到增大到某个值时稳定或者为空，称此时的 δ 取值为饱和点(如图 2 虚线所示)。

4.2.2 δ 的取值和 SVM 算法的分类精度

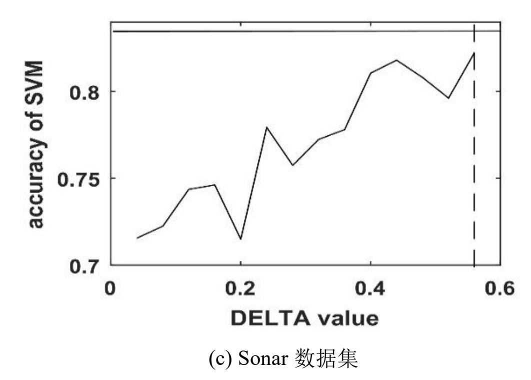
δ 的取值与分类精度的关系如图 3 所示。



(a) Wine 数据集



(b) WDBC 数据集



(c) Sonar 数据集

图 3 δ 取值与 SVM 分类精度之间的关系

分析图 3 中(a)~(c)，横线代表原属性集的分类精度，折线代表不同 δ 取值下对应的分类精度，虚线代表饱和点。以饱和点为基准，将图 3 中(a)~(c)分为前后两部分。根据 4.2.1 节中对饱和点的表述可知：对同一数据集的原属性集而言，前半部分对应着在不同 δ 取值下的约简集合，后半部分对应着原属性集或者属性集合为空。根据实验目的重点分析前半部分。

分析前半部分：首先，3 条折线均呈现上升趋势，且图 3(a)(b)中折线到达某个顶点后稳定。这说明，随着 δ 取值的增大，约简集合的有效性逐渐增加，且在(a)(b)中达到了最大值；其次，各图中饱和点不同，最大精度值和最小精度值之间的差值也不同，特别是在 Sonar 数据集中，这种差值达到了近 10 个百分点。这说明每个数据集的特性是不同的。对于特性各异的数据集，若仅凭经验指定某个 δ 取值，然后用得到的约简结果去证明属性约简算法的效果，这种做法是可以进一步改进的。

4.2.3 实验结论

总体来说，随着δ取值的增大，属性个数增多，分类精度增大，而本文希望的结果是属性个数较少，分类精度较高。针对这种情况，点值式δ取值方式在一定程度上限制了约简算法的实用性。

4.3 点值式与自适应δ取值方法的对比

对点值式δ取值方法设置δ=0.15；对自适应δ取值方法设置$\alpha=0.4$，$\beta=0.6$，且设置δ取值区间为$[0.02,0.4]$，按 0.02 增进，共取得 20 个δ取值。针对 7 个数据集，分别在 SVM 分类器和 1-NN 分类器上对两种方法进行对比。

两种取值方法在 SVM 分类器上的实验结果如表 2 所示。

表 2 F2HARNRS+SVM 的实验结果

数据集	F2HARNRS+SVM					
	点值式			自适应		
	δ	属性个数	分类精度	δ	属性个数	分类精度
Wine	0.15	6	0.9836	0.12	7	0.9945
WDBC	0.15	12	0.9750	0.12	7	0.9743
Sonar	0.15	7	0.7306	0.24	18	0.8256
WPBC	0.15	7	0.6664	0.34	17	0.7705
Ionosphere	0.15	9	0.8832	0.16	9	0.8918
Credit Approval	0.15	13	0.8727	0.02	7	0.8645
German Credit	0.15	13	0.7223	0.12	11	0.7171
平均值		**9.57**	**0.8334**		**10.86**	**0.8626**

两种取值方法在 1-NN 分类器上的实验结果如表 3 所示。

表 3 F2HARNRS+1-NN 的实验结果

数据集	F2HARNRS+1-NN					
	点值式			自适应		
	δ	属性个数	分类精度	δ	属性个数	分类精度
Wine	0.15	6	1.0000	0.20	7	0.9982
WDBC	0.15	12	0.9765	0.10	7	0.9802
Sonar	0.15	7	0.9075	0.32	9	0.9167
WPBC	0.15	7	0.9023	0.12	3	0.9172
Ionosphere	0.15	9	0.9627	0.18	7	0.9754
Credit Approval	0.15	13	0.9319	0.12	13	0.9417
German Credit	0.15	13	0.8757	0.02	7	0.8743
平均值		**9.57**	**0.9367**		**7.57**	**0.9434**

实验结果与数据集和分类器自身的特性及本次实验的实验方案相关，根据以上结果可知：自适应的δ取值方法可以根据实际情况选择较好的δ取值。可以看出，对于同一个分类器而言，在不同的数据集上合适的δ取值是不同的；对于同一个数据集而言，在不同分类器上合适的δ取值也是不同。大体上，相比点值式δ取值方法，通过自适应δ取值方法能找到属性个数更少，而分类精度更高的属性集；或是得到在其中一项接近的情况下，另一项更好的结果。因为结合了实际问题，这种优势是十分明显的，例如在对 WPBC 数据集采用“F2HARNRS+SVM”方式得到的结果中，相比点值式δ取值方法，在自适应的δ取值方法下，分类精度提升了近 11 个百分点，虽然得到约简集合中的属性个数为 17 个，比之多了 10 个，但是原属性集的属性个数是 33 个，该约简集合可以认为是较好的约简集合，反观点值式δ取值方法，虽然得到约简集合中的属性个数更少，但分类精度损失严重，并不能认为是较好的约简集合。可见自适应的δ取值方法能进一步提升算法的实用性。

5 结束语

邻域粗糙集约简算法的效果与邻域大小的δ取值紧密相关。本文对分析了δ取值对算法效果的影响，指出点值式δ取值方法的不足，将对δ取值的评价标准由仅依据人的经验改进为结合实际数据集和分类器的性质，实验证明该方法是可行的。在数据训练阶段求得合适的δ取值，进而得到满足实际需求的属性约简集合，对减轻分类任务的工作量，保持甚至提高分类任务的精度是具有一定意义的。

参考文献：

[1] 王国胤，姚一豫，于洪. 粗糙集理论与应用研究综述 [J]. 计算机学报, 2009, 32 (7): 1229-1246.

[2] Pawlak Z, So-Winski R. Rough set approach to multi-attribute decision analysis [J]. European Journal of Operational Research, 1994, 72 (3): 443-459.

[3] Zadeh LA. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic [J]. Fuzzy Sets & Systems, 1997, 90 (90): 111-127.

[4] Lin TY. Granular Computing on binary relations I: Data mining and neighborhood systems [J]. Rough Sets in Knowledge Discovery, 1998 (2): 165-166.

[5] Hu Q, Yu D, Liu J, Wu C. Neighborhood rough set based heterogeneous feature subset selection [J]. Information Sciences, 2008, 178 (18): 3577–3594.

[6] 王国胤. Rough 集理论与知识获取 [M]. 西安: 西安交通大学出版社, 2001: 147-156.

[7] 胡清华，于达人. 应用粗糙计算 [M]. 北京: 科学出版社, 2012.

[8] Du Y, Hu Q, Zhu P, Ma P. Rule learning for classification based on neighborhood covering reduction [J]. Information Sciences, 2011, 181 (24): 5457-5467.

[9] 刘遵仁，吴耿锋. 基于邻域粗糙模型的高维数据集快速约简算法 [J].

计算机科学, 2012, 39 (10): 268-271.

[10] Liu Y, Huang W, Jiang Y, Zeng Z. Quick attribute reduct algorithm for neighborhood rough set model [J]. Information Sciences, 2014, 271 (7): 65-81.

[11] 段洁，胡清华，张灵均等. 基于邻域粗糙集的多标记分类特征选择算法 [J]. 计算机研究与发展, 2015, 52 (1): 56-65.

[12] Chen H, Li T, Luo C, et al. Dominance-Based Neighborhood Rough Sets and Its Attribute Reduction [M]// Rough Sets and Knowledge Technology. [S. l.] : Springer International Publishing, 2015.

[13] Guo G, Liu Z, Lou C, et al. Improving on a rapid attribute reduction algorithm based on neighborhood rough sets [C]. //Proc of International Conference on Fuzzy Systems and Knowledge Discovery. 2016: 236-240.

[14] Wang C, Shao M, He Q, et al. Feature subset selection based on fuzzy neighborhood rough sets [J]. Knowledge-Based Systems, 2016, 111: 173-179.

[15] Chen Y, Zeng Z, Lu J. Neighborhood rough set reduction with fish swarm algorithm [J]. Soft Computing, 2016: 1-12.

[16] Fan A, Zhao H, Zhu W. Test-cost-sensitive attribute reduction on heterogeneous data for adaptive neighborhood model [J]. Soft Computing, 2016, 20 (12): 4813-4824.